DOCUMENT RESUME

ED 108 607 IR 002 118

TITLE The Use of Simulation in the Design of a Computer-Based Instructional Management System.

INSTITUTION Southwest Regional Laboratory for Educational Research and Development, Los Alamitos, Calif.

REPORT NO SWRL-TN-5-72-12

PUB DATE 17 Mar 72

NOTE 24p.

EDRS PRICE MF-$0.76 HC-$1.58 PLUS POSTAGE

DESCRIPTORS *Computer Programs; Educational Development; *Electronic Data Processing; Electronic Equipment; *Equipment Utilization; Flow Charts; Innovation; Instructional Programs; Instructional Technology; *Management Information Systems; Optical Scanners; Programed Materials; Research Tools; *Simulation; Systems Approach

ABSTRACT

The Southwest Regional Laboratory Instructional Management System (IMS) utilizes a configuration of electronic data processing (EDP) equipment to transform instructional program performance data into instructional program management information. To perform this transformation, SWRL has selected a configuration consisting of on-site optical scanners, low-speed character printers, and a Data Concentrator which communicates with a large central processing system. System operation under each of four IMS-EDP models was simulated using IBM's General Purpose Simulation System (GPSS). The results are reported in terms of device utilization rates, system thruput time and system thruput capacity. They indicate the desirability of multiplexing and variable phasing for optimum system capacity and turnaround time. (Author)

****
* Documents acquired by ERIC include many informal unpublished *
* materials not available from other sources. ERIC makes every effort *
* to obtain the best copy available. nevertheless, items of marginal *
* reproducibility are often encountered and this affects the quality *
* of the microfiche and hardcopy reproductions ERIC makes available *
* via the ERIC Document Reproduction Service (EDRS). EDRS is not *
* responsible for the quality of the original document. Reproductions *
* supplied by EDRS are the best that can be made from the original. *
****

ED 086607

IR

SWRL SOUTHWEST REGIONAL LABORATORY
TECHNICAL NOTE

DATE: March 17, 1972

NO: TN 5-72-12

TITLE: THE USE OF SIMULATION IN THE DESIGN OF A COMPUTER-BASED INSTRUCTIONAL MANAGEMENT SYSTEM

AUTHOR: Burton J. Roffman

ABSTRACT

The SWRL Instructional Management System (IMS) utilizes a configuration of Electronic Data Processing (EDP) equipment to transform instructional program performance data into instructional program management information. To perform this transformation SWRL has selected a configuration consisting of on-site optical scanners, low speed character printers, and a Data Concentrator which communicates with a large central processing system. System operation under each of four IMS-EDP models was simulated using IBM's General Purpose Simulation System (GPSS). The results are reported in terms of device utilization rates, system thruput time and system thruput capacity. They indicate the desirability of multiplexing and variable phasing for optimum system capacity and turnaround time.

U.S. DEPARTMENT OF HEALTH, EDUCATION & WELFARE
NATIONAL INSTITUTE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

IR 008 118

This document is intended for internal staff distribution and use. Permission to reprint or quote from this working document, wholly or in part, should be obtained from SWRL, 11300 La Cienega Boulevard, Inglewood, California, 90304.

THE USE OF SIMULATION IN THE DESIGN OF A COMPUTER-BASED INSTRUCTIONAL MANAGEMENT SYSTEM

The SWRL Instructional Management System (IMS) utilizes a configuration of Electronic Data Processing (EDP) equipment to transform instructional program performance data into instructional program management information and a set of materials and operating procedures which determine how and in what form the data are collected and the information reported. Program performance data include measurements of the performance of pupils as they participate in one or more well defined instructional programs. IMS stores descriptive information about the instructional programs and their participants. The performance data are analyzed, summarized, and combined with this descriptive information to yield instructional program management information. This information is relevant to the requirements of the teachers, parents, principals, district administrators and pupils.

The range of equipment available for IMS includes small, medium and large scale digital computers (central processing units), core memories, disk and drum storage units, magnetic tape drives, paper tape and punched card readers and punches, keyboard input devices, magnetic and optical mark sense readers, character printers, line printers, graphical and optical output devices, telecommunications lines and assorted black box interfaces and connectors. SWRL has initially selected from this range of equipment a configuration consisting of desk top optical mark readers readers, low speed keyboard character printers, and a small scale digital computer with sufficient disk and tape peripherals to function as a data concentrator. The concentrator transmits and receives data over telecommunications lines which connect with the readers and printers, and

with a large scale, multiprogrammed computer system that maintains and executes the main library of IMS processing programs. If the data concentrator unit has the features necessary for multiplexing, then its communications control program may be written to allow simultaneous communications from many readers and/or printers. Otherwise, transactions with each device must be handled separately and sequentially.

One objective of this study was to explore the relative effect of the sequential and the overlapped program logic on the operation of the proposed IMS configuration. However, the operation of the system is equally affected by the timing pattern of the separate processes which the system performs. The basic dichotomy is fixed phasing versus variable phasing. Processes may be constrained to fixed mutually exclusive time intervals within a daily cycle; or, they may occur on-demand, throughout the day, constrained only by equipment availability. Combining these two types of timing patterns with the two alternative program logics yields four IMS-EDP operating models, Fixed Phase Sequential (FPS), Variable Phase Sequential (VPS), Fixed Phase Overlapped (FPO), and Variable Phase Overlapped (VPO). This study investigates the impact on system performance of each of these models in terms of relative device utilization rates, system thruput time, and system thruput capacity.

Performance parameters were estimated using IBM's General Purpose Simulation System (GPSS) program language. The VPS model was defined by a block diagram flowchart according to standard GPSS block types. The diagram was translated into a sequence of GPSS statements, which, when executed under control of the GPSS simulation algorithm, simulate

the activity of IMS with VPS constraints. Various additional statement sequences were merged with the VPS coding to create programs which would simulate the other three models.

The basic VPS system model consists of a variable number of optical scanner input devices and character printer output devices located at participating school sites, a small digital computer data concentrator unit (DCU) placed in a district location, a large scale central processing computer (CPU) in a regional service center, and a two way transmission line (CPLN) between DCU and CPU. All of the scanners have the same operating characteristics as do all of the character printers. It is assumed that the input and output devices will be linked to the DCU by dial-up telecommunications lines, but these lines are not explicitly included in the model because their utilization is identical to the aggregate I/O device utilization.

Transactions which flow through the system undergo a sequence of five processes. The first process is the Input process. One scanner is linked to the DCU. The input data are interpreted from the scan forms, sent to the DCU, coded or concentrated and written onto a magnetic tape file. In the second process the coded contents of this tape are transmitted to a random access disk storage unit which holds the job stream for the central processor. The DCU is linked to the CPU via the telecommunications line (CPLN). This process is labeled the Transmission process. The third process is the Central Computer process. The class test data are merged with data from the instructional program data base and other appropriate sources. Comparisons and analyses are performed

according to programmed algorithms. The resulting information is shaped into a coded file of output report page images which enters the output job queue. In the fourth process this output file is transmitted to the DCU and written onto magnetic tape. The CPU is linked to the DCU via the same telecommunications line used in Process Two. Process Four is called the Receiving process, from the viewpoint of the DCU. The fifth process is Output. The DCU is linked to the character printer at the transaction's originating site. The report page images from the output tape file are decoded, transmitted to the output device and printed as hard copy.

The detail level of the coded model does not include such things as tape usage, disk storage and storage of paper forms, because of the focus on thruput statistics and equipment utilization. Processes are represented by the amount of equipment time required to handle one transaction. These times are coded in five functions, ITIME, TTIME, CTIME, RTIME, and OTIME. One clock unit is defined as one tenth (.1) minute. Transactions begin as the data from one class on one unit of one instructional program enter the model. The entry rate is specified by the GENERATE statement. The listing in Figure 1 shows a GENERATE statement that causes an average interval between transactions of 80 minutes, varying between 40 minutes and 120 minutes. By changing this statement the analyst can simulate different load patterns corresponding to a variety of operating environments.

A transaction enters the model by entering a scanner queue randomly determined by the input device number (IDNUM) function. Changing this

function changes the number of devices in the model and/or the relative number of transactions per device. In the listing IDNUM is set to distribute transactions evenly among twenty scanners. The transaction departs the scanner queue after seizing the appropriate scanner when it becomes available. It then enters the data concentrator input queue (DCIQ), seizes the data concentrator when available, and departs the DCIQ. The scanner and the data concentrator are held for a number of clock units randomly determined by the input time (ITIME) function. In Figure 1 ITIME specifies an average input time of 12 minutes with a range of 8 to 16 minutes. Changing this function changes the simulated speed capabilities of the scanners or the average size of a transaction. This is because transaction size is a derivative of class size and instructional program size. After ITIME has elapsed the scanner and the data concentrator are released and the transaction is buffered. This means it does not proceed directly to the next process and seize the DCU which it has just released. Instead it goes to the end of the queue for that device, allowing transactions which have been generated or advanced during the ITIME interval to compete equally for DCU services.

The Transmission, Receiving and Output processes are coded in a similar manner. Queue statistics are kept for each piece of equipment, aggregately and by process. For example, the DCQ keeps tract of all entries to the data concentrator while the DCOQ monitors only that subset of entries which are seeking Output processing. Equipment is seized according to the priorities of the VPS logic and released after an elapsed time specified by the appropriate function. The Central Computer

process is a partial exception to this pattern. The central processor (CPU) is represented by a storage room entity instead of a unit capacity facility. This allows for the multiprogramming.

As a transaction terminates it is counted and entered into a frequency distribution table according to its thruput time. This table is printed as part of the output statistics for each simulation run. An artificial GENERATE-TERMINATE sequence appears after the TERMINATE statement for the basic model. It acts as a clock and permits the simulation to be run for a specified amount of time rather than a specific number of transactions.

The VPS model was converted into the FPS model by the addition of a time period marking sequence and by the insertion of TEST blocks which prevent transactions from undergoing a process until the time period marker (SAVEVALUE X1) is set for that process. K1 is the code for Input periods, K2 for Output periods, and K3 for Transmission and Receiving. Central Computer processing does not require the services of the DCU and is therefore independent of these time interval restrictions.

The VPO model is also derived directly from the VPS model. A set of computed VARIABLE statements divides the ITIME and OTIME function values into two parts on the basis of the overlap ratio desired. One part applies to the DCU. The other part applies to the input or output device. ADVANCE statements are changed and/or inserted accordingly. The Transmission and Receiving processes are not overlapped because there is only one line between DCU and CPU. Given these FPS and VPO modification provisions it is simply a matter of adding them together to yield the final FPO model, shown in Figure 1.

Figure 1
Card Listing of Fixed Phase Overlapped (FPO) Model

```
      SIMULATE
IPDEV FCN          2(20),0,F
OPDEV FCN          22(20),0,F
MAIN  STORAGE      5
*                                  INPUT DEVICE NUMBER
IDNUM FUNCTION     RN1,C2
0.0,0/1.0,20
*                                  OUTPUT DEVICE NUMBER
ODNUM FUNCTION     RN2,C2
0.0,0/1.0,20
*                                  TIME FOR INPUT PROCESS
ITIME FUNCTION     RN3,C2
0.0,80/1.0,160
*                                  TIME FOR TRANSMISSION PROCESS
TTIME FUNCTION     RN4,C2
0.0,3/1.0,15
*                                  TIME FOR CENTRAL COMPUTER PROCESS
CTIME FUNCTION     RN5,C2
0.0,20/1.0,80
*                                  TIME FOR RECEIVING PROCESS
RTIME FUNCTION     RN6,C2
0.0,10/1.0,20
*                                  TIME FOR OUTPUT PROCESS
OTIME FUNCTION     RN6,C2
0.0,140/1.0,260
*                                  10% OVERLAP DATA CONCENTRATOR TO INPUT DEVICE
    1 VARIABLE     P3/K10
    2 VARIABLE     K9*P3/K10
*                                  10% OVERLAP DATA CONCENTRATOR TO OUTPUT DEVICE
    3 VARIABLE     P11/K10
    4 VARIABLE     K9*P11/K10
*                                  RATE OF INCOMING TRANSACTIONS
      GENERATE     80,40
*                                  ENTER MODEL
      ASSIGN       1,IPDEV
      ASSIGN       1+,FN$IDNUM
      ASSIGN       2,OPDEV
      ASSIGN       2+,FN$ODNUM
      ASSIGN       3,FN$ITIME
      ASSIGN       5,FN$TTIME
      ASSIGN       7,FN$CTIME
      ASSIGN       9,FN$RTIME
      ASSIGN       11,FN$OTIME
*                                  ENTER INPUT DEVICE QUEUE
      QUEUE        P1
      SEIZE        P1
      DEPART       P1
      QUEUE        DCIQ
      QUEUE        DCQ
*                                  WAIT UNTIL INPUT TIME PERIOD
      TEST E       X1,K1
      SEIZE        DCU
      DEPART       DCIQ
      DEPART       DCQ
*                                  INPUT PROCESS
      ADVANCE      V1
      RELEASE      DCU
      ADVANCE      V2
```

```
          RELEASE     P1
          QUEUE       LNQ
*                             ENTER TRANSMISSION LINE QUEUE
          QUEUE       TLNQ
          BUFFER
          SEIZE       CPLN
          DEPART      LNQ
          DEPART      TLNQ
          QUEUE       DCTQ
          QUEUE       DCQ
*                             WAIT UNTIL TRANSMISSION TIME PERIOD
          TEST E      X1,K3
          SEIZE       DCU
          DEPART      DCTQ
          DEPART      DCQ
*                             TRANSMISSION PROCESS
          ADVANCE     P5
          RELEASE     DCU
          RELEASE     CPLN
          QUEUE       MAINQ
          ENTER       MAIN
          DEPART      MAINQ
*                             CENTRAL PROCESSING
          ADVANCE     P7
          LEAVE       MAIN
          QUEUE       LNQ
*                             ENTER RECEIVING LINE QUEUE
          QUEUE       RLNQ
          BUFFER
          SEIZE       CPLN
          DEPART      RLNQ
          DEPART      LNQ
          QUEUE       DCRQ
          QUEUE       DCQ
*                             WAIT UNTIL RECEIVING TIME PERIOD
          TEST E      X1,K3
          SEIZE       DCU
          DEPART      DCRQ
          DEPART      DCQ
*                             RECEIVING PROCESS
          ADVANCE     P9
          RELEASE     DCU
          RELEASE     CPLN
*                             ENTER OUTPUT DEVICE QUEUE
          QUEUE       P2
          BUFFER
          SEIZE       P2
          DEPART      P2
          QUEUE       DCOQ
          QUEUE       DCQ
*                             WAIT UNTIL OUTPUT TIME PERIOD
          TEST E      X1,K2
          SEIZE       DCU
          DEPART      DCOQ
          DEPART      DCQ
*                             OUTPUT PROCESS
          ADVANCE     V3
          RELEASE     DCU
```

```
         ADVANCE     V4
         RELEASE     P2
         TABULATE    THRU
         TERMINATE
         GENERATE    4800,,1
         SAVEVALUE   1,K3
*
         ADVANCE     K150
         SAVEVALUE   1,K2
*
         ADVANCE     K750
         SAVEVALUE   1,K1
*
         ADVANCE     K1050
         SAVEVALUE   1,K3
*
         ADVANCE     K450
         SAVEVALUE   1,K1
*
         ADVANCE     K450
         SAVEVALUE   1,K3
*
         ADVANCE     K450
         SAVEVALUE   1,K2
*
         ADVANCE     K1200
         SAVEVALUE   1,K3
*
         ADVANCE     K300
         TERMINATE
*
         GENERATE    600
         TERMINATE   1
THRU     TABLE       M1,0,150,100
         START       1
         END
```

8:00 - 8:15 TRANSMISSION AND RECEIVING

8:15 - 9:30 INPUT

9:30 - 11:15 OUTPUT

11:15 - 12:00 TRANSMISSION AND RECEIVING

12:00 - 12:45 INPUT

12:45 - 1:30 TRANSMISSION AND RECEIVING

1:30 - 3:30 OUTPUT

3:30 - 4:00 TRANSMISSION AND RECEIVING

LENGTH OF SIMULATION TIME KEEPING

In order to run the simulation programs a number of values and initial conditions must be specified. The number of input and output devices and the values of the five process time functions are common to all four models. Twenty (20) I/O devices prevent bottle necks at particular school sites while allowing a sufficient load to tax the overall processing capacity of the system. Scanner speed is set at five (5) pages per minute. Input transactions average sixty (60) pages (thirty (30) pupils per class, two (2) pages per test). The resulting ITIME function specifies an input process time of twelve (12) minutes plus or minus four (± 4), corresponding to a variance of plus or minus twenty (± 20) pages. Similarly, the character printer speed is set at ten (10) characters per second and output transactions average twelve thousand (12,000) characters (five (5) pages at forty (40) lines per page at sixty (60) characters per line). This yields an Output process time (OTIME) of twenty (20) minutes plus or minus six (± 6), a variance of plus or minus one and one-half (± 1½) pages.

Transmission between DCU and CPU occurs at a rate of two thousand (2,000) bits per second. At eleven (11) bits per character, this is approximately one hundred eighty (180) characters per second. The TTIME function specifies one (1) minute plus or minus one-half (± ½) per transaction, which is derived from sixty (60) pages at one hundred eighty (180) characters per page by one hundred eighty (180) characters per second. A one-half (½) minute variance corresponds to a thirty (30) page variance. This is larger than the twenty (20) page variance implied by ITIME because TTIME is also influenced by start, stop and line failure factors. These

same factors also influence the Receiving process. Thus, RTIME is set at one and one-half (1½) minutes plus or minus one-half (± ½), slightly larger than the sixty seven (67) seconds derived from twelve thousand (12,000) characters by one hundred eighty (180) characters per second. Finally, the CTIME function is set at five (5) minutes plus or minus three (± 3). This figure will be revised when further studies focus on the Central Computer configuration. However, because of the over-riding prominence of the DCU in the present context, the five (5) minute Central Computer processing time acts merely as an additive delay in the thruput time without otherwise affecting system functioning.

Two other initial conditions must be mentioned in addition to these common value specifications. First, in the FPO and VPO models the overlap ratio between DCU and input or output devices is ten percent (10%). For example, if an output process required twenty (20) minutes of printer time it would require two (2) minutes of DCU spread over the twenty (20) minute interval. At this ratio up to ten devices could be handled simultaneously. The second condition relates to the fixed phase models, FPS and FPO. In these models the time period schedule divides an eight (8) hour day as shown in Table 1. The constraints are such that a process may only begin during an interval appropriate to it, although it may run into the interval immediately following.

Each of the models was run under increasing load conditions until the overload point was found. A total of sixteen runs were made. The pertinent statistics from these runs are summarized in Table 2. For comparison purposes, Table 3 selects the statistics which correspond to peak load functioning for each model.

Table 1

Fixed Phasing Schedule

PERIODS DURING WHICH TRANSACTIONS MAY BEGIN A SPECIFIED PROCESS	
8:00 - 8:15	Transmission/Receiving
8:15 - 9:30	Input
9:30 - 11:15	Output
11:15 - 12:00	Transmission/Receiving
12:00 - 12:45	Input
12:45 - 1:30	Transmission/Receiving
1:30 - 3:30	Output
3:30 - 4:00	Transmission/Receiving

The data show that with the given function values under the given conditions the VPS model may be expected to dependably process up to fourteen (14) transactions per day with an average thruput time of approximately two hours. Thruput time is measured from point of entry to the model, which means that delays in seizing an input device are included. Note that in the VPS model thruput time can be reduced to one and two tenths (1.2) hours at a cost of one transaction per day. Conversely, attempting to enter fifteen (15) transactions per day into the system results in huge queue buildups and inflated thruput time with minimal gain in daily thruput. Also, under the overload condition the demand for the CPLN rapidly becomes the primary congestion point.

There is a similar pattern to the data for the FPS model, but on a different scale. At a peak load of fourteen (14) transactions per day

Table 2

Results of Simulation Runs for Four Models

Input Load In Trans./ Day	Percent DCU Util.	Average Wait For CPLN - Minutes	Max. Length Of DCQ	Max. Length Of INQ	Total Thruput	Thruput Per Day	Average Thruput Time - Minutes	Standard Deviation
			Variable Phase Sequential (VPS) - 60 Days					
12 ± 6	84.7	0.0	3	2	720	12.0	55	21.0
13 ± 6½	91.0	0.0	4	3	775	12.9	69	30.3
14 ± 7	97.7	0.0	7	6	833	13.9	122	65.4
15 ± 7½	100.0	883.7	38	73	832	13.9	3680	568.0
			Fixed Phase Sequential (FPS) - 60 Days					
12 ± 6	84.1	25.8	17	14	713	11.9	469	109.5
13 ± 6½	90.9	99.3	22	24	775	12.9	752	161.8
14 ± 7	97.9	199.9	21	26	835	13.9	984	128.9
15 ± 7½	100.0	645.5	38	91	832	13.9	4359	574.4
			Fixed Phase Overlapped (FPO) - 60 Days					
32 ± 16	35.1	47.5	23	28	1900	31.6	344	122.3
48 ± 24	53.2	76.1	29	46	2873	47.9	375	91.6
56 ± 28	62.6	103.2	34	63	3390	56.5	428	92.6
60 ± 30	65.5	276.0	31	132	1763	58.8	759	109.9 *
			Variable Phase Overlapped (VPO) - 60 Days					
68 ± 34	76.0	0.0	5	3	4122	68.7	45	7.7
80 ± 40	89.2	0.1	6	5	4833	80.5	49	8.9
87 ± 44	97.0	18.7	9	9	5263	87.7	59	13.0
96 ± 48	100.0	308.9	38	166	1864	74.5	1281	201.5 *

* 30 days

* 25 days

Table 3

Selected Statistics for Each Model

Model	Thruput* Per Day	Percent DCU Util.	Wait For CPLN - Minutes	Max. Length DCQ	Max. Length LNQ	Average Thruput Time Minutes	Standard Deviation
VPS	13.9	97.7	0.0	7	6	122	65.4
FPS	13.9	97.9	144.4	21	26	984	128.9
FPO	56.5	62.6	103.2	34	63	428	92.6
VPO	87.7	97.0	18.7	9	9	59	13.0

* Unit of thruput equals number of transactions.

the average thruput time is slightly greater than two days with a DCU utilization of ninety seven percent (97%). It requires a reduction of two transactions per day in thruput to reduce the thruput time below one day. One day thruput means overnight turnaround for individual transactions. The contrast between two hour and overnight turnaround is the major difference between the VPS and FPS models. It is also notable that, with the FPS model, even under the lighter load conditions, there is considerable delay in obtaining the CPLN to perform the Transmission and Receiving processes. This is due to the fixed phasing which automatically introduces a certain amount of delay and waste.

A sharp contrast in average thruput time also exists between the two overlapped models. At fifty six (56) transactions per day the FPO model exhibits a thruput time of approximately seven (7) hours. This corresponds to a DCU utilization of only sixty two percent (62%).

However, the size of the queue for CPLN and the associated delay time indicate that this level is just below overload. This is confirmed by the statistics for an input load of sixty (60) transactions per day. The model only processes fifty nine (59) transactions and there is rapid escalation in queue buildups and time delays. In the case of the VPO model the average thruput time is less than one hour at eighty (80) transactions per day and eighty nine percent (89%) DCU utilization. This is more thruput in considerably less time than the FPO model. Also the average waiting time for CPLN is negligible as with the VPS model.

It is possible that an adjustment of the timing schedule shown in Table 1 would improve the performance of the fixed phase models. However, initial attempts to follow this suggestion have been unsuccessful. As the data indicates, it appears that overlapping or multiplexing of input and output devices is critical to the transactional capacity of a computer-based IMS, with the increase over sequential models being at least a factor of five. At the same time, any turnaround time significantly less than overnight seems to depend on the elimination of fixed phasing or mutually exclusive time periods for separate processes in the IMS operating procedures. The size parameters of each installation impose different performance requirements on the proposed system. IMS designers may plan accordingly.

This study is valuable for the planning and design information it has evolved, and because it demonstrates how systems simulation may be used to determine the best operating procedures for a given assembly of

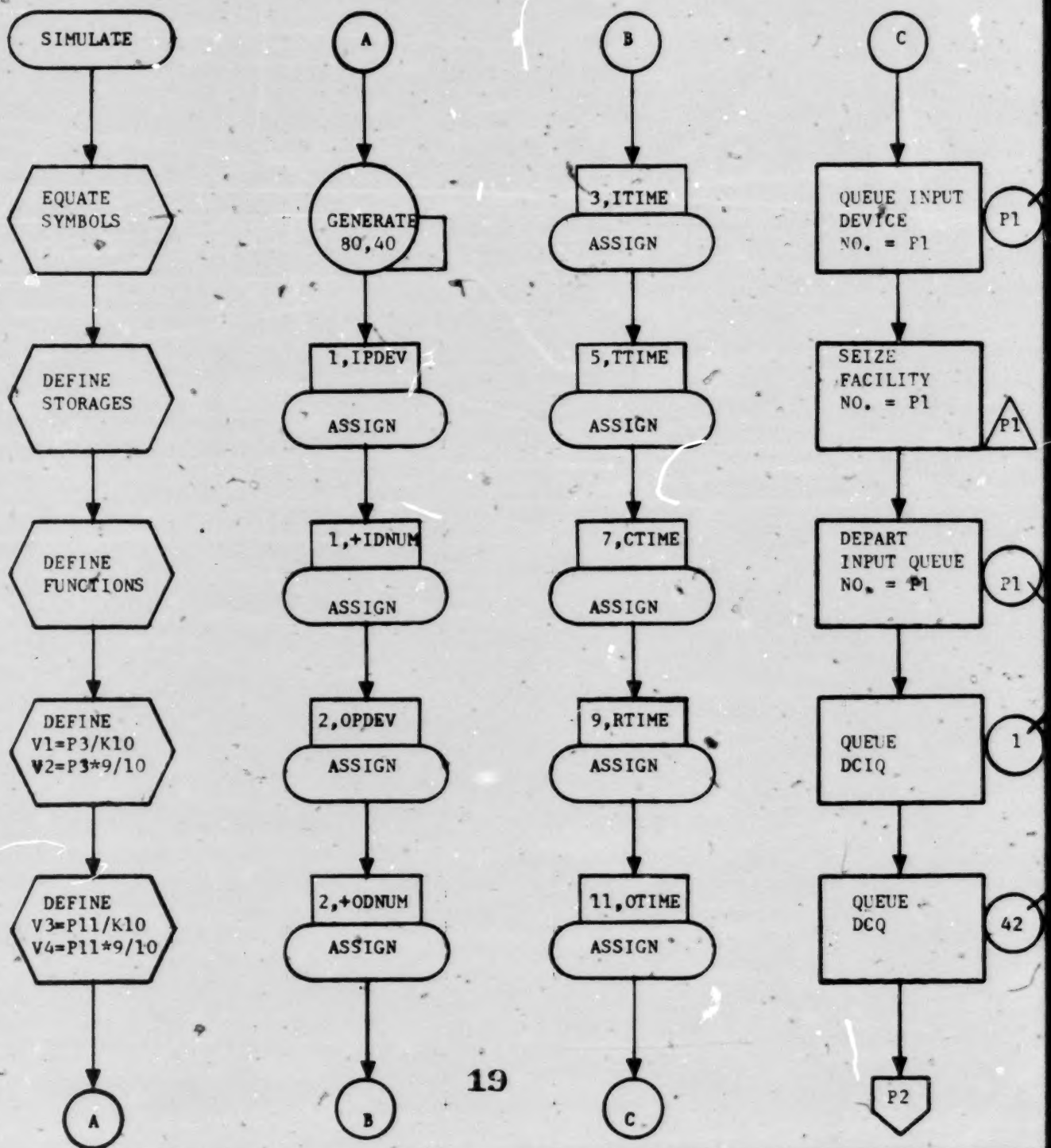
equipment in a given environment. It provides an operational model for a computer-based instructional management system and shows how this model may be coded for simulation purposes. It relates the coding to the aspects of the real system which are represented and shows how necessary function values may be derived. The type of statistical results which a simulation yields are exhibited and discussed in context. The completed models maintain an on-going facility for the evaluation of newly developed technology relevant to the modeled system. For example, the impact of a faster input scanner could be easily studied by changing the values of the ITIME function. Useful information might also be obtained by expanding the present models and increasing their level of detail.

APPENDIX

FLOWCHART LAYOUT FORM

PROGRAMMER: _____ DATE: _____

PROGRAM ID: _____ PAGE _____

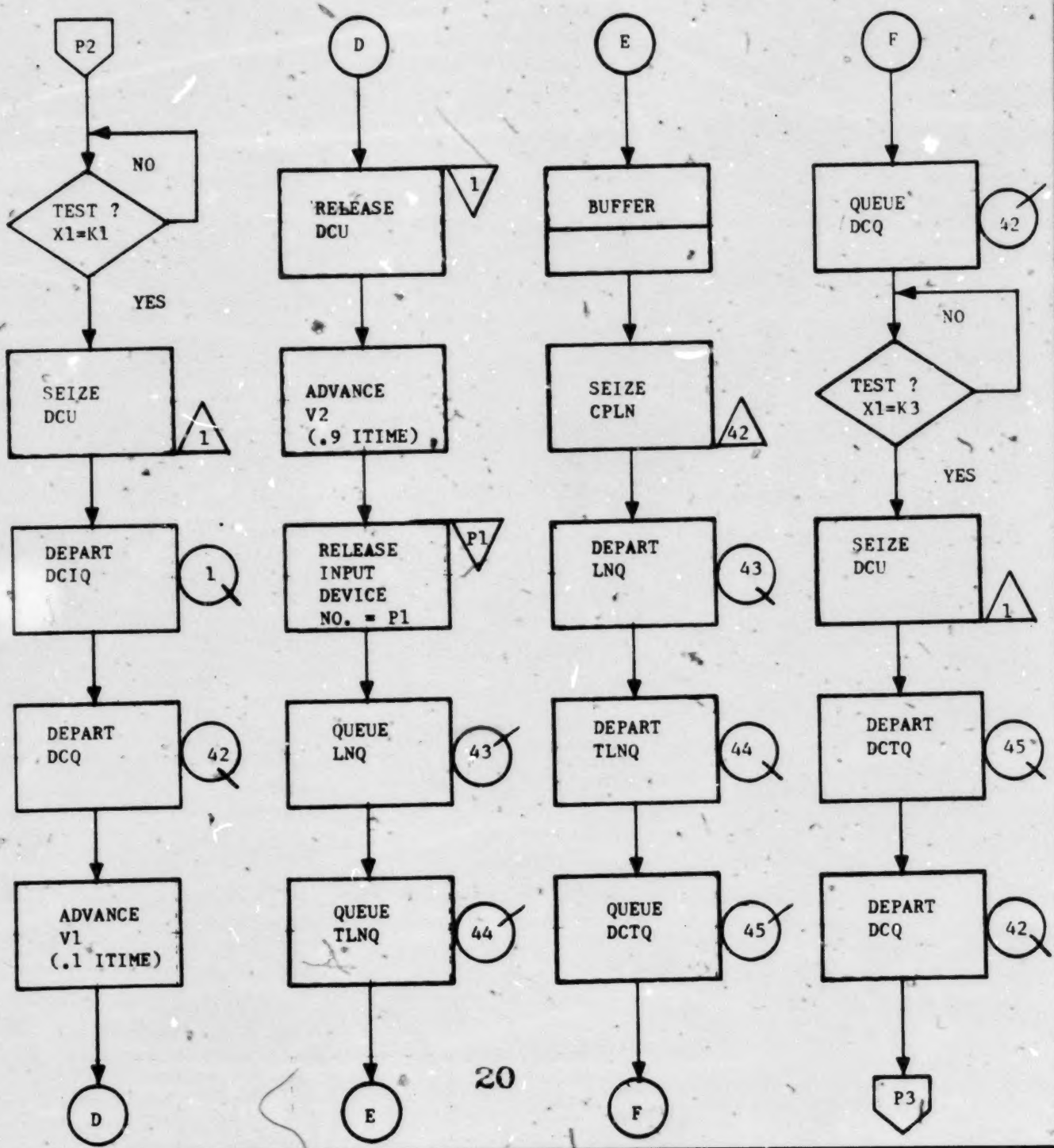


SWRI

FLOWCHART LAYOUT FORM

PROGRAMMER: _____ DATE: _____

PROGRAM ID: _____ PAGE _____

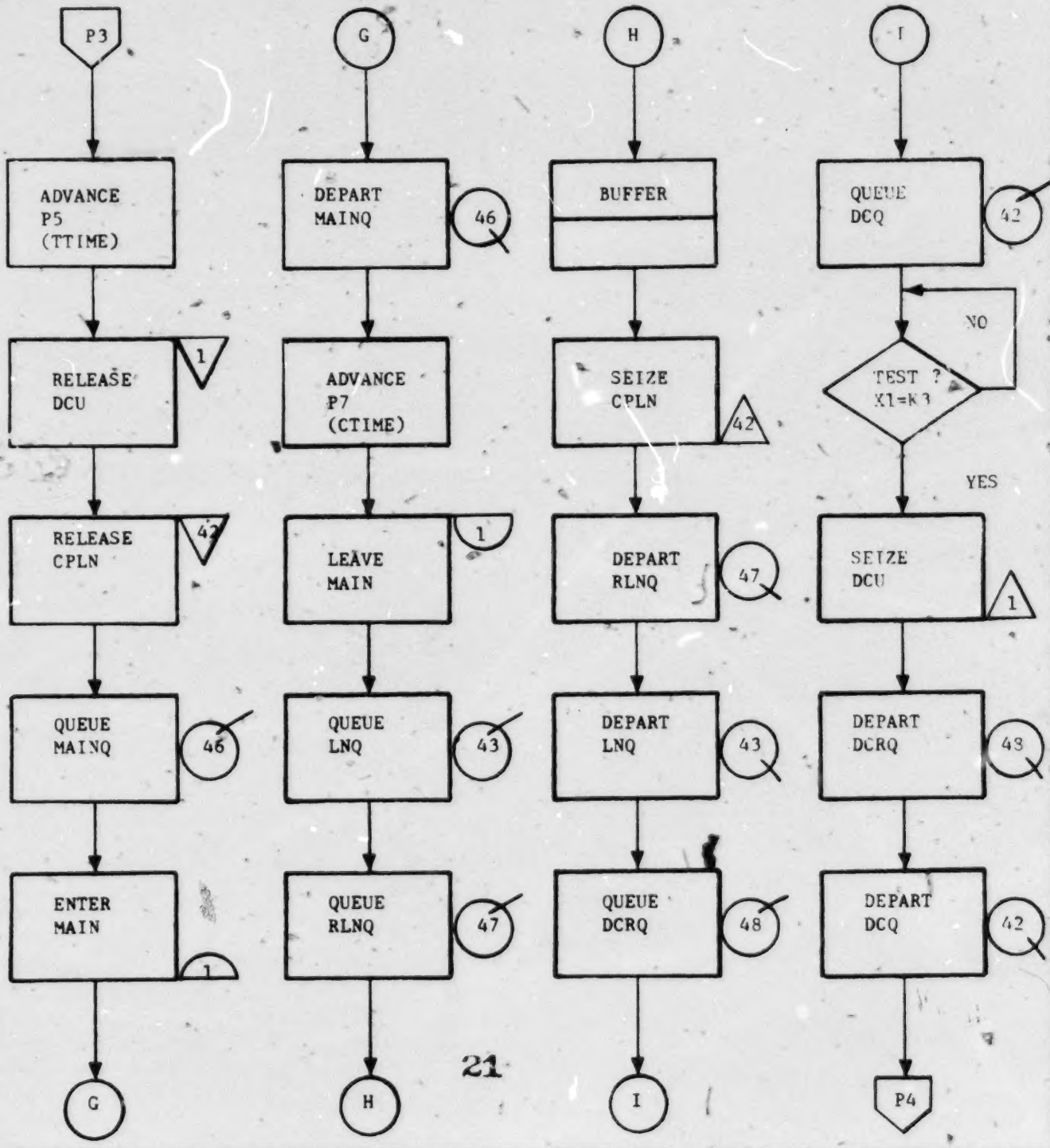


SWRI

FLOWCHART LAYOUT FORM

PROGRAMMER: _____ DATE: _____

PROGRAM ID: _____ PAGE _____

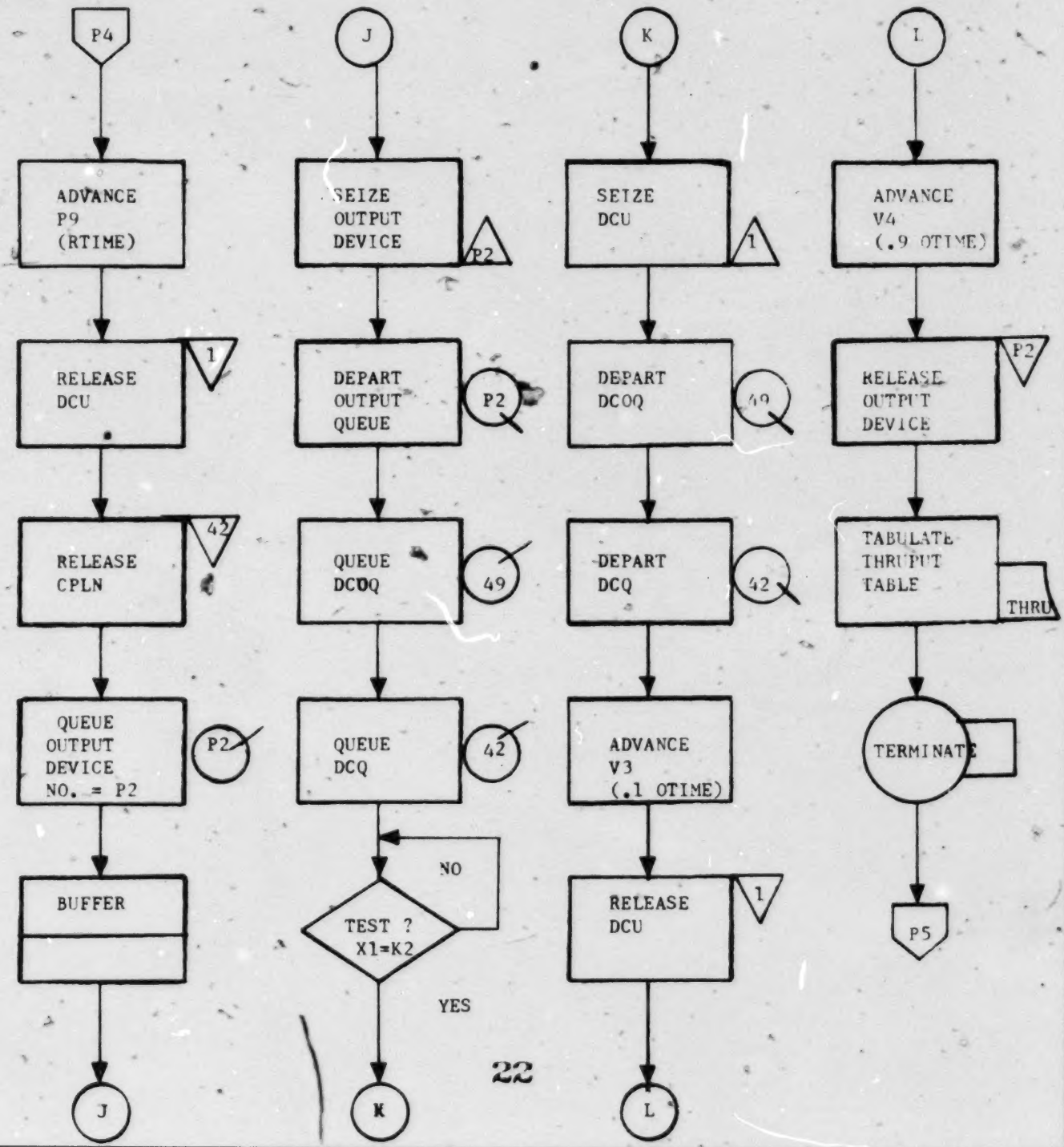


SWRI

FLOWCHART LAYOUT FORM

PROGRAMMER: _____ DATE: _____

PROGRAM ID: _____ PAGE _____

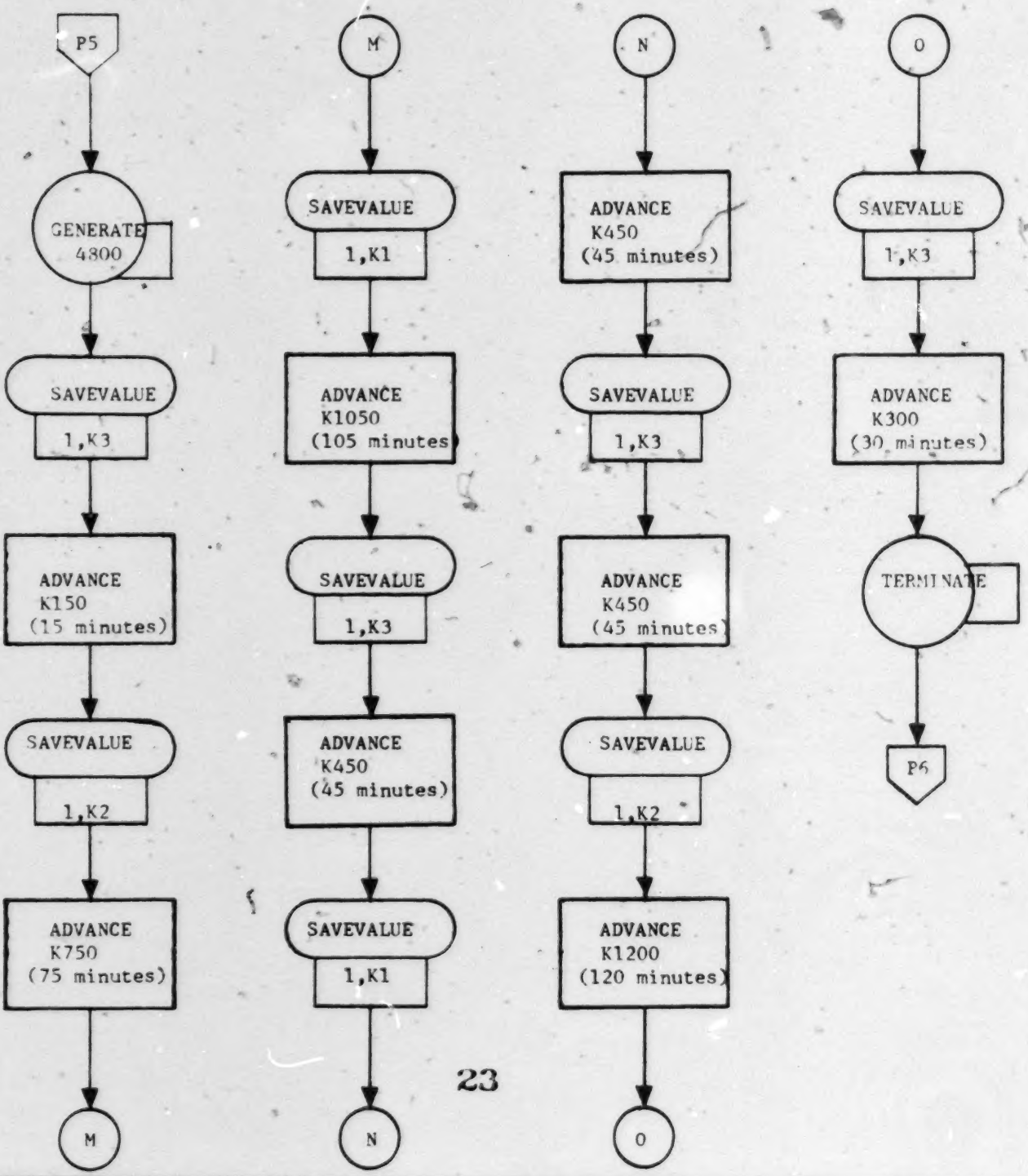


SWRI

FLOWCHART LAYOUT FORM

PROGRAMMER: _____ DATE:

PROGRAM ID: _____ PAGE



SWRI

FLOWCHART LAYOUT FORM

PROGRAMMER: _____ DATE: _____

PROGRAM ID: _____ PAGE _____